

# Söpört felületek felismerése a digitális alakzatrekonstrukcióban

TDK dolgozat

**Kovács István**

BME TTK, Matematika BSc, III. évfolyam



Témavezető:

**Várady Tamás**

tudományos főmunkatárs

BME IIT

2012

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>5</b>
1.1. Digitális alakzatrekonstrukció és a kutatás célja . . . . .	5
1.2. Háromszöghálók . . . . .	6
1.3. Szegmentáció, felületek osztályozása . . . . .	6
1.4. Söpört felületek . . . . .	7
1.5. A feladat nehézségei . . . . .	8
<b>2. Lokális felületi paraméterek becslése</b>	<b>9</b>
2.1. Definíciók . . . . .	9
2.2. Skaláris görbületi jellemzők becslése . . . . .	10
2.2.1. Síkszerűség . . . . .	10
2.2.2. Gauss-görbület . . . . .	10
2.2.3. Átlaggörbület . . . . .	11
2.2.4. Főgörbületek . . . . .	11
2.3. Görbületi irányok becslése . . . . .	11
<b>3. Vektormező egységesítés, simítás</b>	<b>13</b>
3.1. Görbület-stabilitás szűrő . . . . .	14
3.2. Régiók osztályozása . . . . .	14
3.3. Vektormező szinkronizációja, gyenge görbületű részek simítása . . . . .	15
<b>4. Felületi görbék követése (tracing)</b>	<b>17</b>
4.1. Az algoritmus . . . . .	17
4.2. Zárt görbék . . . . .	19
4.2.1. A megállási feltételek . . . . .	19
4.2.2. Javító görbe . . . . .	20
4.2.3. Görbék átlagolása . . . . .	20
<b>5. Görbületi rácsok</b>	<b>21</b>
5.1. Görbületarányos paraméterezés . . . . .	21
5.2. Naiv algoritmus . . . . .	22
5.3. Lyukak kikerülése . . . . .	22

<b>6. A profil- és vezérgörbe meghatározása</b>	<b>24</b>
6.1. Görbületi vonalak . . . . .	24
6.2. Elsődleges illesztés . . . . .	26
6.3. Pontos illesztés, az ICP eljárás . . . . .	26
6.4. A görbék átlagolása . . . . .	27
6.5. A vezérgörbe meghatározása . . . . .	28
<b>7. Az eredmények kiértékelése</b>	<b>29</b>
7.1. A profilgörbe vizsgálata . . . . .	29
7.2. Távolság térkép . . . . .	29
<b>8. Megvalósítás, keretrendszer</b>	<b>30</b>
<b>9. Összefoglalás, további célok</b>	<b>31</b>

## Kivonat

A modern háromdimenziós szkennerek segítségével nagy felbontással tudunk pontokat mintavételezni létező tárgyak felületéről. A digitális alakzatrekonstrukció (vagy mérnöki visszafejtés) célja, hogy mért pontfelhők alapján digitális reprezentációt készítsünk olyan esetekben, ahol ilyen számítógépes modell nem létezik. Az eljárásnak számos mérnöki alkalmazása van, többek között régi alkatrészek újragyártásában és gyártás-minőségellenőrzésben, de gyakran használják az orvostudományban is testreszabott gyógyászati eszközök (például protézisek, műfogak) készítéséhez.

A mért pontfelhők feldolgozása igen komplex feladat. Először a pontokból háromszöghálót kell készíteni, amely alapján megbecsüljük a felületek különböző geometriai jellemzőit és a felületet különálló tartományokra osztjuk (szegmentáljuk). Ezek után a szegmentált tartományokat automatikusan osztályoznunk kell aszerint, hogy milyen típusú matematikai felülettel lehet ezeket legjobban közelíteni.

Jelen kutatás egy speciális felület-osztályozási feladatra összpontosít, söpört felületek felismerésére és rekonstrukciójára. A söpört felületek egy állandó síkbeli profil eltolásával keletkeznek egy másik görbe (az ún. gerinc- vagy vezérgörbe) mentén, és jelentős szerepet játszanak különböző mérnöki alkalmazásokban. Az általunk fejlesztett algoritmussal eldönthető, hogy egy mért adatokból álló felület jól közelíthető-e egy söpört felülettel, és ha igen, azt rekonstruálni is tudjuk. Az algoritmus lokálisan becsült görbületi jellemzők vizsgálatán alapszik. Görbületi rácsok létrehozását követően előállítjuk az optimális söprő síkot, a profilt és vezérgörbét. A legjobban illeszkedő matematikai reprezentáció megtalálása után vizsgálnunk kell, hogy az illesztés mennyire pontos, és elfogadható-e az algoritmus eredménye.

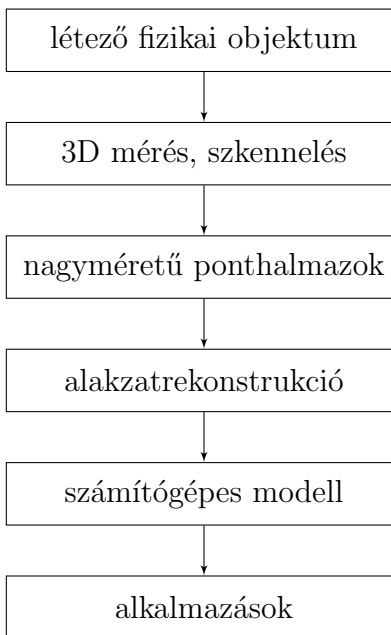
A feladatot nagymértékben nehezíti a mérésekből származó zajosság, ami miatt szükséges a vektormezők javítása. Egy másik nehéz probléma, hogy általában a felület tartomány nem mindig teljes a szomszédos felületekből származó áthatások következményeként. Így kisebb-nagyobb belső tartományok hiányozhatnak, amiket a görbületi rács létrehozásakor ki kell kerülnünk, vagy át kell ugranunk. A fentiek illusztrálják, hogy a söpört felületek felismerése és rekonstrukciója mért adatokból alapján valóban összetett és nehéz feladat.

A dolgozatban bemutatjuk az algoritmus felépítését, majd valós adatokból származó modelleken szemléltetjük a feladat nehézségét és az implementáció eredményét.

# 1. Bevezető

## 1.1. Digitális alakzatrekonstrukció és a kutatás célja

A háromdimenziós szkennerek elterjedésével a mért adatok feldolgozása egyre fontosabb kérdéssé válik. Tipikus feladat például, hogy egy létező objektumból mérések alapján digitális (CAD) modellt készítsenek, ezzel megkönnyítve régi alkatrészek újragyártását és módosítását, illetve felgyorsítva a tervezés-gyártás ciklust. A felületi méréseknek az orvostudományban is számos alkalmazása van, gondoljunk csak a testreszabott térdprotézisekre vagy a digitális fogászatra. Ezt a folyamatot digitális alakzatrekonstrukciónak vagy mérnöki visszafejtésnek nevezzük [3]. Az egyik legelterjedtebb ezzel foglalkozó szoftver a Geomagic Studio (<http://geomagic.com>), ahol a szkenneléstől a kész CAD modellig a teljes alakzatrekonstrukció végrehajtható.



1. ábra. A digitális alakzatrekonstrukció folyamata

A szkennerek által létrehozott pontfelhőkből háromszöghálót kell készítenünk, majd ezt különböző geometriai jellemzők alapján logikailag összefüggő tartományokra bontjuk (szegmentáljuk). A szegmentáció után az egyes tartományokról eldöntjük, hogy az milyen matematikai felülettel reprezentálhatók legjobban.

Ezen kutatás célja, hogy egy mért adatokból származó háromszöghálóról meghatározzuk, hogy az jól közelíthető-e állandó profilú söpört felülettel, és ha igen meghatározzuk a matematikai reprezentációt is. A söpört felületek egy állandó síkbeli profil eltolásával keletkeznek egy másik görbe (az ún. gerinc- vagy vezérgörbe) mentén. Ehhez a feladathoz a háromszögháló lokálisan becsült görbületi jellemzőit használjuk fel, és a felületen létrehozott görbületi rács segítségével határozzuk meg a profilt majd a vezérgörbét is. Az algoritmus a következő fő lépésekből áll:

1. Skaláris és vektoriális görbületi jellemzők megbecslése a háromszöghálón
2. Görbületi rács létrehozása
3. Görbületi vonalak fedésbe hozása és átlagolása (profil meghatározása)
4. A becsült profil visszahelyezésével a vezérgörbe meghatározása
5. A teljes reprezentáció meghatározása és annak vizsgálata

Az algoritmust egy általam fejlesztett prototípus-implementációs környezetben teszteltük, több számítógéppel generált, illetve mért adatokból származó háromszöghálón is. Az ismert algoritmusok mindegyike megvalósult, amit a program futása közben készített képekkel illusztrálunk.

A dolgozatban ismertetjük az algoritmus működését, a felmerülő problémákat és az azokra adott megoldásokat.

## 1.2. Háromszöghálók

A háromdimenziós szkennerek nagy méretű pontfelhőket mintavételeznek a tárgyak felületéről. Ahhoz, hogy ezt a pontfelhőt felületként értelmezni tudjuk, a pontokból háromszöghálót kell készíteni. Erre van több hatékony és jól ismert algoritmus is. A dolgozatban feltételezzük, hogy a beérkező adathalmaz háromszögháló, az alakzatrekonstrukció ezen részével nem foglalkozunk.

Ahhoz, hogy bármilyen felületi jellemzőt meg tudjunk becsülni, szükségünk van arra, hogy a háromszöghálót könnyen be tudjuk járni. Ehhez „fél-él” adatstruktúrában tároljuk a háromszöghálót, ami nagyon leegyszerűsítve azt jelenti, hogy olyan irányított gráfként reprezentáljuk a modellt, amiben minden pont között oda-vissza fut él. Minden élhez tartozik egy háromszögtartomány is, a háromszögtartományok körül pedig ciklikusan irányítódnak az élek. Ez a CAD világban az egyik legelterjedtebb adatstruktúra, amivel a háromszögháló nagyon hatékonyan bejárható.

A háromszöghálón, mint gráfon a pontoknak meghatározhatjuk a szomszédait, amik nem feltétlen közvetlen szomszédok. Egy  $p$  pont  $n$ -ed rendű szomszédait a továbbiakban SZOMSZÉD( $p, n$ )-el jelöljük.

## 1.3. Szegmentáció, felületek osztályozása

A háromszöghálót különböző geometriai jellemzők alapján különálló tartományokra kell osztanunk, amikről külön-külön eldönthetjük, hogy milyen matematikai felülettel közelíthetőek. A szegmentáció célja, hogy a felületet elsődleges tartományokra, és az őket határoló elválasztó tartományokra ossza. Ez általában lokális görbületi jellemzők alapján történik, ahol egybefüggőnek tekintünk egy tartományt, ha az adott jellemzők határértéken belül maradnak. Szintén figyelembe vehetjük a felületen található hirtelen változásokat, esetleges lekerekítéseket, ahol egyből felismerhetjük, hogy azok tartományokat határolnak (pl. Morse szegmentáció). Ahogy később láthatjuk, felhasználhatunk még görbületi rácsokat is szegmentációs feladatokhoz, amelyek adott esetben jobb eredményt adhatnak, mint a skaláris görbületi jellemzőkkel dolgozó algoritmusok.

A szegmentáció után minden elsődleges tartományról el kell döntenünk, hogy az milyen matematikai felülettel közelíthető. Ezt úgy tehetjük meg, hogy először egyszerűbb felületekkel próbáljuk azt megilleszteni, és ha az illesztés egy megadott határértéken belül jó, megkaptuk a tartomány egy jó reprezentációját. Ha nem sikerült az illesztés, egyre bonyolultabb felületekkel próbálkozunk. Amennyiben az illesztés nem járt sikerrel, próbálkozhatunk újbóli szegmentációval, majd a kisebb tartományon újból az illesztéssel. Ha ez sem hatásos, azaz ha a tartomány alapvető felületekkel nem közelíthető, a tartományt szabadformájú felületnek hívjuk.

A szegmentáció során a következő alapvető felülettípusokba próbáljuk meg besorolni az elsődleges tartományokat:

1. Sík
2. Kihúzott (extruded) felület (henger; általános profil)
3. Ferdén kihúzott (drafted) felület (kúp; általános profil)
4. Forgásfelület
5. *Állandó profilú söpört (swept) felület*
6. Változó profilú söpört (lofted) felület
7. Nyílt szabadformájú felület
8. Zárt szabadformájú felület

A jelen projektben megvalósított algoritmus célja a mért adatokból származó állandó profilú söpört felületek optimális matematikai reprezentációjának megtalálása. Amennyiben mást nem mondunk, feltételezzük, hogy a söpört felület állandó profilú.

A dolgozatban a szegmentációs módszereket nem részletezzük, a söpört felületek felismerése esetében feltételezzük, hogy a bejövő modell már egy szegmentált tartomány. A háromszöghálón megengedhetünk lyukakat, ami egy valós alakzatrekonstrukciós probléma esetében szinte mindig előfordul.

## 1.4. Söpört felületek

A söpört felületek egy állandó síkbeli profil eltolásával keletkeznek egy másik görbe (az ún. gerinc- vagy vezérgörbe) mentén. A vezérgörbe mentén eltolt söprő sík koordinátarendszerének orientációját pedig egy rotációs függvény határozza meg.

Legyen  $\phi : \mathbb{R} \rightarrow \mathbb{R}^3$  folytonos paraméterezett síkbeli ( $z = 0$ ) profilgörbe,  $\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$  differenciálható, önmagát nem metsző vezérgörbe,  $r : \mathbb{R} \rightarrow \mathbb{R}^3$  rotációs függvény, ami a vezérgörbére merőleges, azaz  $\langle \dot{\gamma}(u), r(u) \rangle = 0$ . Legyen  $M_r(u)$  az a forgatás-mátrix ami az  $(1, 0, 0)^\top$  vektort az  $r(u)$  irányába, a  $(0, 0, 1)^\top$  vektort pedig a  $\dot{\gamma}(u)$  irányába forgatja. Ekkor az ezek által definiált söpört felület:

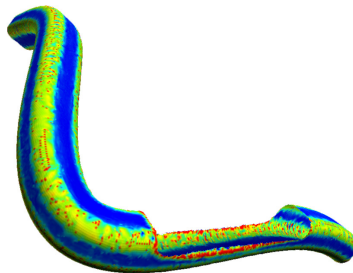
$$s(u, v) = \gamma(u) + M_r(u)\phi(v)$$

Ennek a felülettípusnak nagyon nagy szerepe van a mérnöki alkalmazásokban, nagyon sok alkatrész vagy annak egy része eleve egy söpört felületből származik, amiknek a felismerése mért adatokból nem egyszerű feladat, főleg ha a felület nem teljes, azaz például lyukas.

## 1.5. A feladat nehézségei

A feladatunk eldönteni, hogy egy modell adott tartománya közelíthető-e söpört felülettel. Ezt úgy kell megtennünk, hogy a felületről semmilyen információnk nincsen, a profilt és a vezérgörbét is nekünk kell meghatározni. Ehhez a felület görbületi jellemzőit használjuk fel, amiknek a számítása nem egyszerű feladat főleg nagyobb méretű zajos háromszögháló esetében. Egy szkennelt modellnél számíthatunk több százezer pontból álló háromszöghálóra is, ezért gyors és hatékony algoritmusokat kell alkalmaznunk. A megbecsült görbületi skalár- és vektormezők alapján szükség van görbületi vonalak létrehozására, ami egy differenciálegyenlet diszkrét megoldására vezethető vissza. A görbületi vonalak alapján el kell készíteni a modell egy konzisztens görbületi rácsát, amelyet felhasználva már felismerhetjük egy söpört felület jellemzőit.

A mért adatokból származó adatok gyakran zajosak, ami a lokális görbületi jellemzők becslésekor is megjelenik (2. ábra). Ez nagyban megnehezíti a feladatot, hiszen az algoritmus alapját adják a becült görbületi jellemzők. Ezt a problémát a görbületi skalár- vagy vektormezők javításával, illetve a túlságosan zajos régiók átugrásával vagy kikerülésével oldottuk meg.



2. ábra. Egy zajos és lyukas mért adatokból származó felület

A legtöbb mérnöki alkatrész esetén a söpört felület csak az alakzat egy része, ezért a szegmentációból származó tartomány hiányos, azaz lyukas. Ha az egész modell ténylegesen egy söpört felületből származik akkor is számíthatunk arra, hogy a végénél nem a vezérgörbére merőleges síkkal vágják azt el, így a felület végénél a söprő sík nem metszi ki a teljes profilt. Az algoritmusunk görbületi rácsok felhasználásával dolgozik, így ha a görbületi rácsot lyukas háromszöghálókon is létre tudjuk hozni, – például ha kikerüljük, vagy átugorjuk a felületen található lyukakat – az algoritmus célravezető. A megvalósított algoritmus képes kezelni az olyan lyukakkal rendelkező söpört felületeket, ahol nincs két lyuk egymással szemben, pontosabban amikben a söprő sík mindig összefüggő profildarabot metsz ki a felületből.



## 2. Lokális felületi paraméterek becslése

A háromszöghálók jellegének felismeréshez szükség van arra, hogy a differenciálgeometriai lokális felületi jellemzőket meg tudjuk becsülni. Ezek lehetnek skaláris vagy vektoriális jellemzők.

Itt fontos kérdés, hogy a becslés mennyire legyen lokális, és hogy hogyan választjuk ki az approximáció alapjául szolgáló háromszögeket. Szokás adott távolságú környezetből gyűjteni az információt, illetve a háromszögelés alapján is értelmezhetjük a távolságot, azaz hogy hány éllel érhető el az adott pont. Mi ez utóbbit használjuk. Egyszerű geometriai jellemzők esetén az adott pontnak csak a közvetlen szomszédjai alapján becsüljük meg az adott jellemzőt.

A következőkben használt szükséges jelölések: legyen  $p$  a vizsgált pont helyvektora,  $q_1, q_2, \dots$  pedig a szomszédos pontok körüljárási irány szerinti sorrendben.

- Az élek:  $e_i = q_i - p$
- Háromszögek normálvektorai:  $n_i = \frac{e_i \times e_{i+1}}{\|e_i \times e_{i+1}\|}$
- $\alpha_i = \angle(e_i, e_{i+1})$
- $\beta_i = \angle(n_i, n_{i+1})$
- Háromszögek területe:  $A_i = \frac{\|e_i \times e_{i+1}\|}{2}$
- $A(p) = \sum_i A_i$
- Szöghiány:  $\delta(p) = 2\pi - \sum_i \alpha_i$

### 2.1. Definíciók

Most összefoglaljuk a későbbiekben szükséges differenciálgeometriai definíciókat, állításokat. A most következő fogalmakat csak áttekintő jelleggel ismertetjük, a pontos definíciók, bizonyítások felépítése meghaladná a dolgozat kereteit.

Legyen  $g : \mathbb{R} \rightarrow \mathbb{R}^3$  paraméterezett görbe. Ekkor  $\kappa : \mathbb{R} \rightarrow \mathbb{R}$  görbülete:

$$\kappa(u) = \frac{|\dot{r}(u) \times \ddot{r}(u)|}{|\dot{r}(u)|^3}$$

Legyen  $r : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  paraméterezett felület, és legyen  $t_p$  a felület egy  $r(p)$  pontbeli érintővektora,  $n_p$  normálvektora. A  $t_p$  érintővektor és az  $n_p$  felületi normális által kifeszített sík a felületből egy görbét metsz ki, melynek görbülete  $r(p)$ -ben a  $t_p$  irányhoz tartozó normálgörbület abszolútértéke.

A normálgörbület definíciójából adódóan elég csak az  $r(p)$ -beli érintősíkban egy  $r(p)$  középpontú egységkörrel választani a  $t_p$  érintővektort. Így a normálgörbületek egy egységkörön értelmezett folytonos függvényt határoznak meg, amely felveszi a minimumát és maximumát is. Ezeket nevezzük  $\kappa_{min}(p)$  és  $\kappa_{max}(p)$  főgörbületeknek, a hozzájuk tartozó  $t_{min}$  és  $t_{max}$  érintővektorokat pedig főgörbületi irányoknak. A felület bármely pontjában van két egymásra merőleges főirány. Kivételt

képeznek, az ún. umbilikus pontok, amikor a két főgörbület azonos, és a görbület minden irányban ugyanaz, így nincs értelme minimumról és maximumról beszélni. Az umbilikus pontok lokális gömb- vagy síkszerű felületeken fordulnak elő.

## 2.2. Skaláris görbületi jellemzők becslése

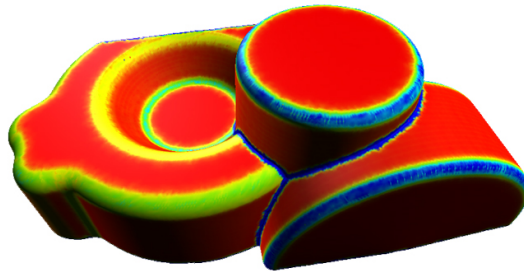
A háromszögelt felületen különféle lokális görbületi jellemzők becslésére van szükségünk. Ezekre léteznek ismert algoritmusok [2], [9], amelyeket a következőekben ismertetni fogunk.

A használt görbületi jellemzők:

- Síkszerűség
- Gauss- más néven szorzatgörbület
- Átlaggörbület
- $\kappa_{min}, \kappa_{max}$  főgörbületek

### 2.2.1. Síkszerűség

A síkszerűséget egy adott pont környezetében levő pontokon becsült normálvektorok szórása adja meg [4]. Azaz legyenek a környező normálvektorok  $n_1, n_2, \dots, n_k$ , legyen ezeknek a lenormált átlaga  $n_a$ . Az átlagtól való átlagos eltérés:  $\sum_i \frac{|n_i - n_a|}{k}$  adja meg a síkszerűséget, ugyanis ha síkszerű a felület, ott a környező normálvektorok nagyjából egy irányba mutatnak, ezért ez az érték kicsi lesz.

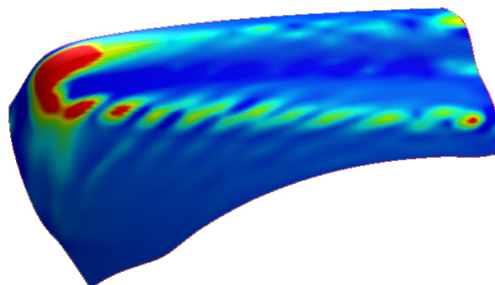


3. ábra. Síkszerűségi térkép

### 2.2.2. Gauss-görbület

A Gauss görbület a főgörbületek szorzatából adódik, azaz  $G = \kappa_{min}\kappa_{max}$ . Ekkor a következő diszkrét becslés adható:

$$G(p) = \frac{\delta(p)}{\frac{1}{3}A(p)}$$

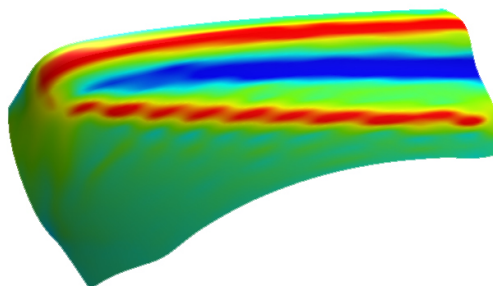


4. ábra. Gauss görbületi térkép

### 2.2.3. Átlaggörbület

Az átlaggörbület a főgörbületek átlaga, azaz  $H = \frac{\kappa_{min} + \kappa_{max}}{2}$ . A diszkrét becslése pedig:

$$H(p) = \frac{\frac{1}{4} \sum_i \beta_i \|e_i\|}{\frac{1}{3} A(p)}$$



5. ábra. Átlaggörbületi térkép

### 2.2.4. Főgörbületek

Ezek alapján a főgörbületek értéke triviálisan adódik.

$$\kappa_{min,max} = H \pm \sqrt{H^2 - G}$$

## 2.3. Görbületi irányok becslése

A továbbiakban a főirányok által meghatározott vektormezővel fogunk foglalkozni. Elegendő csak a  $\kappa_{min}$  főirány meghatározása hiszen a  $\kappa_{max}$  erre merőleges. Ezen főirány meghatározása hengerillesztéssel történik. Vegyük a  $\kappa_{min}(p)$  meghatározásához a  $p$  pont ( $n$ -ed rendű) szomszédaihoz tartozó normálvektorokat a Gauss-gömbön. Mivel a vektorok mind egység hosszúak, ezt úgy érjük

el, hogy egy síkot illesztünk úgy, hogy a sík átmenjen az origón, és a vektoroktól való négyzetes távolság minimális legyen. Legyenek a környező normálvektorok  $n_1, n_2, \dots$  a megfelelő  $x_i, y_i, z_i$  koordinátákkal. Legyen a keresett sík normálvektora  $n$ ,  $N_x, N_y, N_z$  koordinátákkal. Ekkor egy  $n_i$  vektor síktól való előjeles távolsága  $\left\langle \frac{n}{\|n\|}, n_i \right\rangle$ , tehát minimalizálandó a  $\sum_i \left\langle \frac{n}{\|n\|}, n_i \right\rangle^2$ . Ezt átalakítva, együtthatómátrixos alakban írva

$$\begin{aligned} \sum_i \left\langle \frac{n}{\|n\|}, n_i \right\rangle^2 &= \sum_i \frac{1}{N_x^2 + N_y^2 + N_z^2} (N_x x_i + N_y y_i + N_z z_i)^2 = \\ &= \sum_i \frac{1}{N_x^2 + N_y^2 + N_z^2} [N_x, N_y, N_z] \begin{bmatrix} x_i^2 & x_i y_i & x_i z_i \\ x_i y_i & y_i^2 & y_i z_i \\ x_i z_i & y_i z_i & z_i^2 \end{bmatrix} \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = \frac{\langle n, An \rangle}{\langle n, n \rangle} \end{aligned}$$

ahol

$$A = \sum_i \begin{bmatrix} x_i^2 & x_i y_i & x_i z_i \\ x_i y_i & y_i^2 & y_i z_i \\ x_i z_i & y_i z_i & z_i^2 \end{bmatrix}$$

Az  $\frac{\langle n, An \rangle}{\langle n, n \rangle}$  pontosan a Rayleigh-hányados, amiről tudjuk, hogy a minimuma az  $A$  mátrix legkisebb abszolútértékű sajátértéke, és ezt a hozzá tartozó sajátvektoránál veszi fel. Egyébként látszik az is, hogy  $A$  nagyon hasonlít a  $n_i$  pontok végpontjaihoz mint egységnyi tömegpontokhoz tartozó tehetetlenségi tenzorhoz, konkrétan

$$\Theta = I - A$$

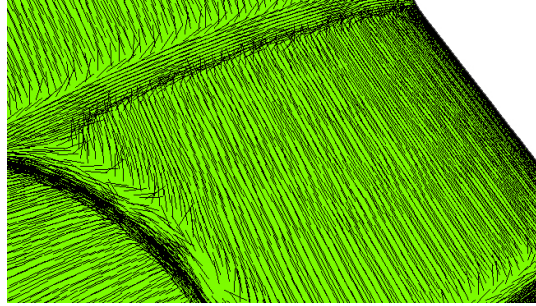
Keressük tehát  $A$  legkisebb abszolútértékű sajátértékéhez tartozó sajátvektorát. Ennek megtalálásához az inverz iterációs módszert használjuk, ami a következő. Legyen  $M$  egy tetszőleges mátrix, sajátértékei abszolút nagyság szerinti sorrendben  $\lambda_1, \lambda_2, \lambda_3$ , hozzájuk tartozó sajátvektorok  $e_1, e_2, e_3$ , és tegyük fel hogy ezek lineárisan függetlenek (jelen esetben mindenképpen azok, hiszen  $A$  sajátvektorai ortonormált bázist alkotnak). Az iteráció kiindulási vektora pedig  $v$  aminek az  $e_1, e_2, e_3$  bázisban való felírása  $v = \gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3$ . Ekkor

$$M^n v = \lambda_1^n \gamma_1 e_1 + \lambda_2^n \gamma_2 e_2 + \lambda_3^n \gamma_3 e_3$$

amiben ha  $n$  nagy,  $\lambda_3^n$  fog dominálni, így a vektor nagyjából  $e_3$  irányába fog mutatni, ez bőven elég nekünk, hiszen csak a vektor irányára vagyunk kíváncsiak. Tapasztalat szerint már három iterációs lépés célravezető.

Mivel  $A$  legkisebb abszolútértékű sajátértéke  $A^{-1}$  legnagyobb abszolútértékű sajátértéke, és a hozzá tartozó sajátvektor ugyanaz, ezért  $A^{-1}$ -re alkalmazzuk az iterációs módszert. Arra érdemes figyelni, hogy a kiindulási  $v$  vektor nehogy az  $e_3$ -ra merőleges lineáris altérbe essen, hiszen ekkor az iteráció nem vezet célra. A konkrét megvalósításban ezzel tényleg külön kell foglalkozni, a részleteket most hanyagoljuk. Az algoritmus eredményét a 6. ábrán láthatjuk.

A másik főirány pedig  $e_3$ -ra merőleges vektor lesz, az éppen vizsgált pont érintősíkjaiban.



6. ábra. Főgörbületi irányok egy hengerszerű felületen

### 3. Vektormező egységesítés, simítás

Előfordulhat, hogy a görbületi irányok minimálisból a maximális görbületi irányba váltanak át (továbbiakban min-max váltás). Ekkor a normál vonalkövetési eljárásokkal megáll az algoritmus, hiszen a min-max váltások határán a görbületi irányok becslése nagyon bizonytalan. Ennek egy lehetséges megoldása, ha detektáljuk a min-max váltás határát, majd a vonalkövetés esetén átugorjuk azt. Egy másik lehetséges megoldás, hogy szintén felismerjük a régiók határát, és eldöntjük, hogy egy adott régióban a minimális vagy a maximális görbületi vektormezőt kell alkalmaznunk. Ez utóbbi megoldást fogjuk a következőkben részletezni, az eredményt a 8. ábra szemlélteti.

Egy konzisztens görbületi rácsháló létrehozásához szükség van egy egységes vektormezőre, amin stabilan működik a vonalkövetés, nincs benne zajos régió és nincsenek min-max váltások. Amennyiben a felületen nincsenek umbilikus pontok a min-max váltásokon kívül, a vektormező létrehozható, és így a felület jól paraméterezhető. Mostantól feltesszük, hogy a vizsgált felület ilyen.

Gyenge görbületű tartományoknak nevezzük azokat a régiókat, ahol a vektormezőnek nagy az irány szerinti szórása, azaz elegendően közel esnek egymással nagy szöget bezáró vektorok. A nem gyenge görbületű tartományokat erős görbületűnek nevezzük. A közelség és a szög itt paraméterként szerepel, ez tipikusan a háromszöghálón 1 vagy 2 sugarú legyező és a szögeltérés nagyobb, mint  $30^\circ$ . A definícióból adódóan a min-max váltások határa mindig gyenge görbületű tartomány. Továbbá feltesszük azt is, hogy lokálisan, két egymáshoz közel levő vektor állása azonos, azaz az általuk bezárt szög kisebb mint  $90^\circ$ .

Az algoritmus működése vázlatosan a következőkből áll:

1. Gyenge görbületű tartományok detektálása
2. Szélességi bejárással a min-max váltások felismerése a gyenge görbületű határok mentén, régiók osztályozása, hogy minimális vagy maximális vektormezőt használjunk
3. Vektormező szinkronizálása a detektált régiók alapján
4. Vektormező simítása a tartományok határán és a gyenge görbületű részekben

A további fejezetekben ezeket sorban megvizsgáljuk, és bemutatjuk a feladatokra adott megoldásokat.

### 3.1. Görbület-stabilitás szűrő

A görbület-stabilitás szűrő megkeresi a felület azon pontjait, ahol a becsült görbületi irányok lokálisan nagyjából egy irányba állnak. Ezeket nevezzük erős görbületű pontoknak. Minden más esetben a pontot gyenge görbületű pontnak nevezzük. Az ilyen pontokban a görbületi vonalkövetés elakadna, vagy rossz eredményt adna. Adott  $p$  csúcsponttól szeretnénk eldönteni, hogy stabil-e ott a görbület egy adott  $\varepsilon$  tűréshatáron belül. A  $\kappa_{min}$  görbületi irányokat meghatározó egységvektor mező. Az algoritmus eldönti, hogy egy  $p$  pont erős görbületű-e (7. ábra).

---

#### Algoritmus 1 GÖRBÜLETSTABILITÁS( $p, n, \varepsilon$ )

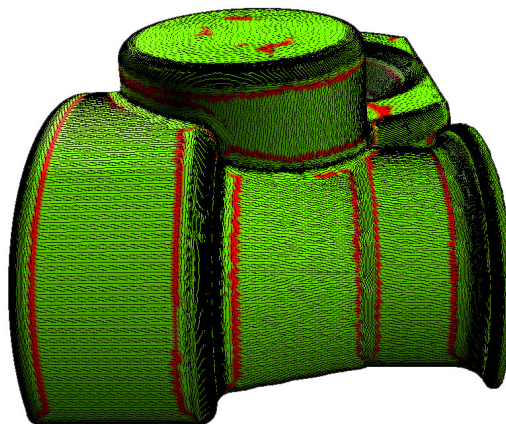
---

```

for all  $u, v \in \text{SZOMSZÉD}(p, n)$  do
  if  $\langle \kappa_{min}(u), \kappa_{min}(v) \rangle < 1 - \varepsilon$  then
    return HAMIS
  end if
end for
return IGAZ

```

---



7. ábra. Gyenge görbületi térkép

### 3.2. Régiók osztályozása

A gyenge görbületi régiók a min-max váltások határa mentén a felületet szegmentálják. Ezeknek a régióknak a határain el kell döntenünk, hogy valójában min-max váltás történik, vagy esetleg csak zajosság vagy egyéb görbületi bizonytalanság (pl. hirtelen merőleges törés) miatt detektáltuk a határukat gyenge görbületűnek. Ezek után a tartományokat két osztályba soroljuk azzal a tulaj-

donsággal, hogy két szomszédos tartományon a vektormező a határok mentén egymásra merőleges legyen.

Az algoritmus gyakorlati megvalósításában egy megfelelő kiindulópontból szélességi bejárással csoportosítottuk a régiókat. A szélességi bejárással az éppen aktuálisan vizsgált pont szomszédai egy részén az algoritmus már lefutott, ami feltétlenül szükséges a működéshez. Az algoritmus csak a fejzet elején feltételezett ideális felületeken működik (sőt csak ott van értelme), azaz ha a felületen nincsenek umbilikus pontok.

Adott  $p$  csúcsról szeretnénk eldönteni, hogy melyik típusú (minimális  $V_{min}$ , maximális  $V_{max}$ ) tartományhoz tartozik, adott egy gyenge görbület szűrő, egy  $\kappa_{min}$ ,  $\kappa_{max}$  egymásra merőleges görbületi vektormező, és a  $p_0$  kiindulópont. Az algoritmus egy adott pontjáig bejárt pontjairól a  $\kappa_{min-max}$  a  $V_{min}$  beli pontokhoz a  $\kappa_{min}$ -t, a  $V_{max}$  pontjaihoz pedig a  $\kappa_{max}$ -ot rendeli. Az algoritmus a gyenge görbületű pontokat nem osztályozza, a  $\kappa_{min-max}$  ezeken (egyenlőre) nincs értelmezve. Az algoritmus által egy adott pontig bejárt pontok legyenek  $V$  halmaz elemei. Az  $\hat{A}TLAG(A)$  az  $A$  halmaz elemeinek átlagát jelöli.

$$\kappa_{min-max}(p) = \begin{cases} \kappa_{min}(p) & \text{ha } p \in V_{min}(p) \\ \kappa_{max}(p) & \text{ha } p \in V_{max}(p) \end{cases}$$

---

**Algoritmus 2** MINMAXBEOSZTÁS( $p_0, n, n_A, \varepsilon$ )

---

```

 $V_{min} \leftarrow p_0$ 
for all  $p \in \text{SZÉLES}(p_0)$  do
  if GÖRBÜLETSTABILITÁS( $p, n, \varepsilon$ ) then
     $V \leftarrow p$ 
     $A = \{\kappa_{min-max}(q) : q \in \text{SZOMSZÉD}(p, n_A), \text{GÖRBÜLETSTABILITÁS}(q, n, \varepsilon), q \in V\}$ 
     $v_{avr} = \hat{A}TLAG(A)$ 
     $v_{avr} = \frac{v_{avr}}{\|v_{avr}\|}$ 
    if  $|\langle v_{avr}, \kappa_{min}(p) \rangle| > |\langle v_{avr}, \kappa_{max}(p) \rangle|$  then
       $V_{min} \leftarrow p$ 
    else
       $V_{max} \leftarrow p$ 
    end if
  end if
end for

```

---

### 3.3. Vektormező szinkronizációja, gyenge görbületű részek simítása

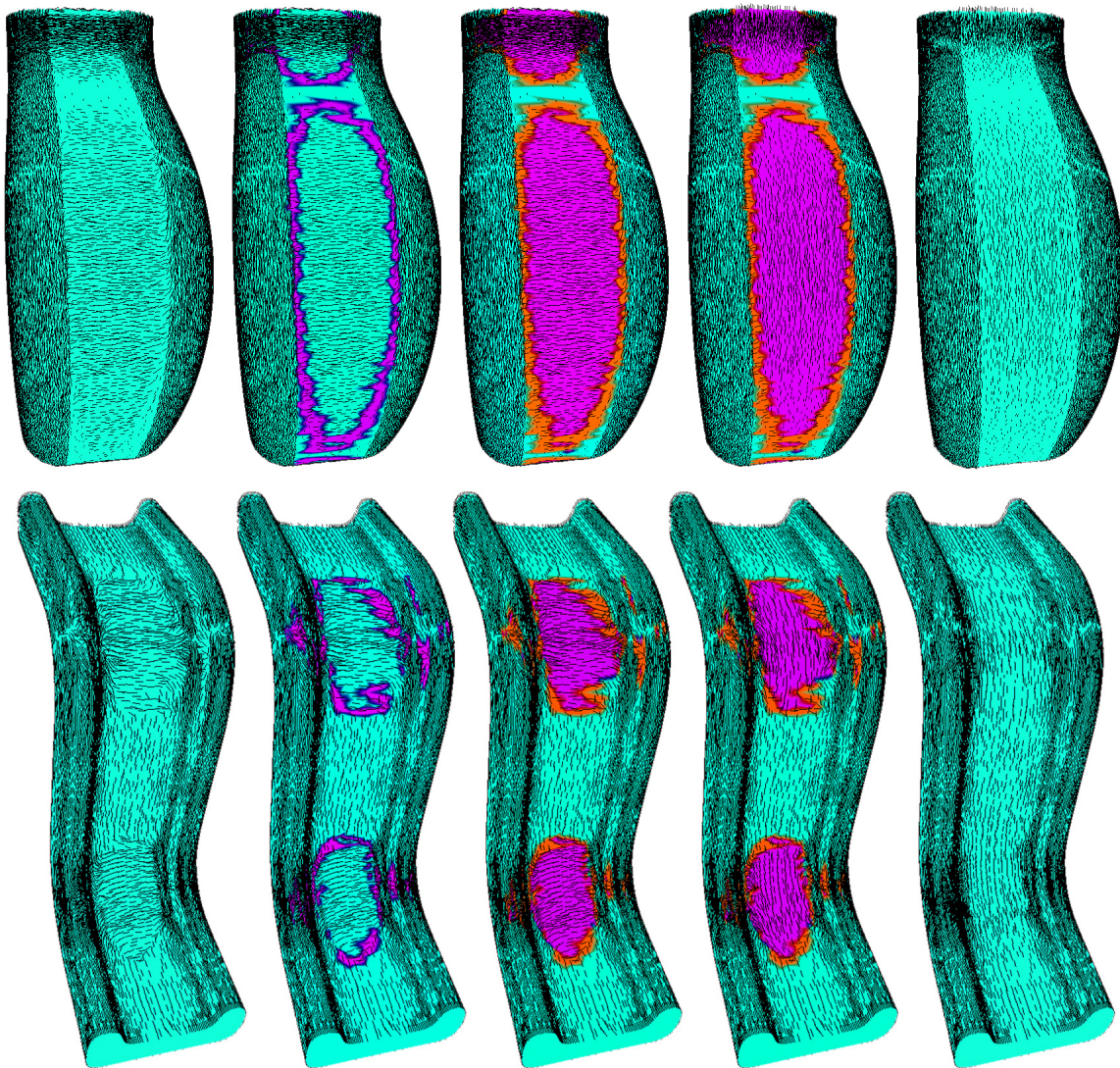
Az előzőekben megadott algoritmus kimenete alapján már szinkronizálható a vektormező, azaz a  $\kappa_{min-max}$  a minimális régiókhoz tartozó pontokhoz a  $\kappa_{min}$ -t, a maximális régiókhoz tartozókhoz pedig a  $\kappa_{max}$ -ot rendeli. A  $\kappa_{min-max}$ -ot a gyenge görbületű részekeken még értelmeztük. Ezekre a pontokra vesszük a szomszédos nem gyenge görbületű pontok  $\kappa_{min-max}$ -ainak átlagát.

$$A(p) = \{\kappa_{min-max}(q) : q \in \text{SZOMSZÉD}(p, n_A), \text{GÖRBÜLETSTABILITÁS}(q, n)\}$$

$$\kappa_{min-max}(p) = \begin{cases} \kappa_{min}(p) & \text{ha } p \in V_{min}(p) \\ \kappa_{max}(p) & \text{ha } p \in V_{max}(p) \\ \text{ÁTLAG}(A(p)) & \text{egyébként} \end{cases}$$

Megjegyezzük, hogy ez utóbbi lépést érdemes lehet iterálni is, azaz az új  $\kappa_{min-max}$  vektormezőn még egyszer lefuttatni a gyenge görbületű tartományok detektálását, és újra elvégezni az átlagolást a gyenge görbületű pontokra.

Ezek után elkészült az egységes  $\kappa_{min-max}$  vektormező, ami már alkalmas a vonalkövetésre.



8. ábra. A vektormező egységesítő algoritmus lépései: 1. A modell a lokálisan becsült vektormezővel, 2. Gyenge görbületű tartományok detektálása, 3. Régiók osztályozása, 4. Vektormező szinkronizálása, 5. A fennmaradt részek simítása, a végeredmény



## 4. Felületi görbék követése (tracing)

Adott egy háromszögháló, és rajta egy vektormező. A cél találni egy felületen futó görbét, amelynek deriváltja mindig a vektormező irányába mutat. Ezt egyszerű, felületen futó törtvonalak létrehozásával oldottuk meg.

Görbületi vonalnak nevezünk egy olyan felületen futó görbét, aminek az érintője minden pontban görbületi irány, így ha a vonalkövetéshez a görbületi vektormezőt használunk, a kapott görbék görbületi vonalak lesznek. Ez egy diszkrét megoldása a  $\dot{\varphi} = \kappa_{min}$  vagy  $\dot{\varphi} = \kappa_{max}$  differenciálegyenletnek. Ez alapján Runge-Kutta közelítéssel kaphatunk felületen futó törtvonalakat. A kapott görbék mindig paraméterezettek, vagy egyszerű ívhossz szerint, vagy egy görbületi skalármező segítségével. Erre azért van szükség, hogy később a görbületi rácsokat görbülettel súlyozottan tudjuk létrehozni.

Az egyszerű vonalkövető algoritmus a háromszöghálón élről élre lép, minden pontban a vektormező irányába teszi meg a következő lépést. A Runge-Kutta módszer minden lépésben az új pozíció helyett visszaugrunk az új szakasz feléhez, majd onnan folytatjuk a vonalkövetést.

A vonalkövetés az általános felhasználhatóság érdekében jól paraméterezhető, és a megállási feltételek is részletesen testreszabhatóak.

### 4.1. Az algoritmus

A vonalkövetést a következő mennyiségekkel jellemezhetjük:

- Kiindulási pozíció ( $p_0$  pont a háromszöghálón, lehet háromszög közepében is)
- Vektormező ( $\kappa : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ )
- Skalármező ( $\mu : \mathbb{R}^3 \rightarrow \mathbb{R}$ )
- Kiindulási irány, illetve kétirányúság
- Megállási feltételek

A vonalkövetés végeredményeként kapunk egy  $G : \mathbb{R} \rightarrow \mathbb{R}^3$  paraméterezett törtvonalat.  $G$  pontjai  $V(G) = \{p_0, p_1, \dots, p_n\}$ , élei  $E(G) = \{e_1, e_2, \dots, e_n\}$ , ahol az  $e_k$  a  $p_{k-1}$  és  $p_k$  között fut.

Egyszerű vonalkövetés esetében élről élre lépünk, adott  $p_i$  pozícióból, egy  $l_i$  élről, vagy egy  $h_i$  háromszögből. A következő,  $p_{i+1}$  pont meghatározása a következőképpen történik:

---

#### Algoritmus 3 LÉPÉSEGYSZERŰ( $p_i, \kappa$ )

---

A  $p_i$  pontban becsült felületi normális  $n_i$ . Vesszük  $l_i$  szomszédos éleit (vagy  $h_i$ -t határoló eléket) a háromszöghálón, majd ezeket elmetszük egy  $p_i$ -n átmenő  $n_i \times \kappa(p_i)$  normálvektorú síkkal. Az így kapott metszéspontok legyenek  $q_1, q_2, \dots, q_k$ . Ezek közül válasszuk ki azt, amire  $\angle(q_j - p_i, l_i)$  maximális. Ez lesz  $p_{i+1}$ .

---

A lépést Runge-Kutta módszerrel finomíthatjuk. Ezzel egy lépésben két új pontot kapunk.

**Algoritmus 4** LÉPÉSRUNGEKUTTA( $p_i, \kappa$ )

Határozzuk meg a következő  $q$  pontot a LÉPÉSEGYSZERŰ algoritmus segítségével. A következő pont  $p_{i+1} = \frac{p_i+q}{2}$ . Ezután a  $p_{i+2}$  pontot  $p_{i+1}$ -ből kiindulva a LÉPÉSEGYSZERŰ módszer segítségével kaphatjuk meg.

A vonalkövetést bizonyos esetekben nincs értelme folytatni, például ha zajos, gyenge görbületű régióba érünk, vagy éles töréshez érünk. Speciális feladatokhoz később bevezetünk más megállási feltételeket is, például, ha egy másik görbébe ütközünk, vagy a kezdőpont túl közel kerül a végpont-hoz. A legtöbbször használt megállási feltételek a következők:

**Algoritmus 5** MEGÁLLÁS( $p_i$ )

MEGÁLLÁSHOSSZ( $h$ ): Megállunk, ha  $\sum_{j=1}^i \text{hossz}(e_j) > h$ .

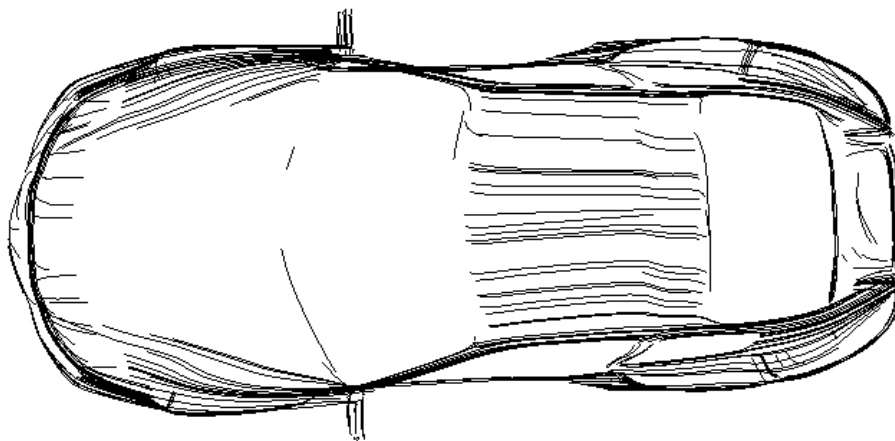
MEGÁLLÁSLÉPÉSHOSSZ( $n$ ): Megállunk, ha  $i > n$ .

MEGÁLLÁSGYENGE: Megállunk, ha  $p_i$  gyenge görbületű.

MEGÁLLÁSUMBILIKUS: Megállunk, ha  $p_i$  umbilikus pont.

MEGÁLLÁSSZÖG( $\alpha$ ): Megállunk, ha  $\angle(e_{i-1}, e_i) > \alpha$ .

A megállási feltételek együtt legyenek MEGÁLLÁS( $p$ ). A LÉPÉS( $p, \kappa$ ) pedig  $p$  pontból kiindulva újabb pont LÉPÉSEGYSZERŰ vagy LÉPÉSRUNGEKUTTA módszerrel,  $c$  korrekciós tag a görbület szerinti paraméterezéshez, aminek később látjuk az értelmét. Az algoritmus eredménye egy  $G$  felületen futó törtvonal,  $V(G)$  pontokkal, és  $E(G)$  élekkel reprezentálva. Egy autómódel  $\kappa_{min}$  vektormezőjén követett görbületi vonalakat a 9. ábra szemlélteti.



9. ábra. Göribületi vonalak egy autómódel

**Algoritmus 6** TRACE( $p_0, \kappa, \mu, c$ )

---

```

 $p = p_0$ 
while  $\neg$ MEGÁLLÁS( $p$ ) do
   $V(G) \leftarrow p$ 
   $p_n = \text{LÉPÉS}(p, \kappa)$ 
   $e = (p, p_n)$ 
   $\text{hossz}(e) = |p_n - p|(c + \mu(\frac{p_n+p}{2}))$ 
   $E(G) \leftarrow e$ 
   $p = p_n$ 
end while

```

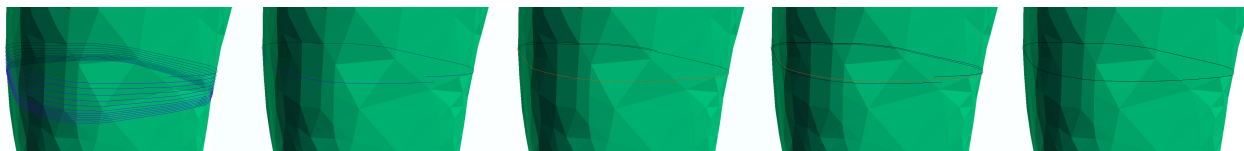
---

## 4.2. Zárt görbék

Egy topológiailag hengerszerű felületen szeretnénk létrehozni egy zárt görbületi vonalat, viszont egy  $p$  pontból elindított vonalkövetés nem biztos, hogy  $p$ -be jut vissza a felület és a vektormező zajossága, és a becslések bizonytalansága miatt. Azt viszont felismerhetjük, hogy a görbe  $p$ -nek egy megadott  $\varepsilon$  sugarú környezetébe jutott-e vissza. Ha igen, azt feltételezhetjük, hogy a görbe zárt, és egy másik követett görbe segítségével becsülhetünk egy, az eredeti görbékhez jól illeszkedő görbületi vonalat. Az algoritmus lépéseit a 10. ábra illusztrálja.

Az algoritmus működése vázlatosan:

1. Felismerni görbekövetés közben, ha visszaértünk a kiindulópont kis környezetébe
2. Tovább folytatjuk a görbekövetést addig, amíg a körbe kezdő és végpontja ( $s$  és  $t$ ) a lehető legközelebb van
3. Kétirányú követés indítása hasonló megállási feltétellel  $\frac{s+t}{2}$ -ből
4. A két görbületi vonal egyesítése megfelelő súlyozással



10. ábra. 1. Egy felületen a generált követési görbe csavarvonal szerűen halad, 2. A vonalkövetés megállt, a kezdőpont és a végpont elegendően közel került, ( $G$ ) 3. Javító görbe ( $G_j$ ), 4. Az eredeti és a javító görbe, 5. Az átlagolt görbe ( $G_a$ ), a végeredmény

### 4.2.1. A megállási feltételek

Ha a követési görbe végpontja elegendően közel ér a kiindulópontához megállunk, és megállapítjuk, hogy a görbe bezárható, a megállási feltétel a következő:

Adott egy görbekövetési algoritmus, amely minden  $i$ -edik lépésben ad egy  $p_i$  pontot, ezek határozzák meg a görbét. Legyen  $\varepsilon$  előírt távolság paraméter. A kiindulópont pedig legyen  $p_0$ .

---

MEGÁLLÁSZÁRT( $p_0, p_i, \varepsilon$ ): Megállunk, ha  $p_0 \neq p_i$ ,  $|p_0 - p_i| < \varepsilon$  és  $|p_0 - p_i|$  minimális olyan értelemben, hogy  $|p_0 - p_i| < |p_0 - p_{i\pm 1}|$ .

---

Amennyiben egy kétirányú görbét indítunk, a megállási feltétel hasonló. Egy pontból indítunk két ellentétes irányú vonalkövetést, a két görbe pontjai  $p_i, q_i$ .

---

MEGÁLLÁSDUPLAZÁRT( $q_i, p_i, \varepsilon$ ): Megállunk, ha  $q_0 \neq q_i, p_0 \neq p_i$ ,  $|q_i - p_i| < \varepsilon$  és  $|q_i - p_i|$  minimális olyan értelemben, hogy  $|q_i - p_i| < |q_{i\pm 1} - p_{i\pm 1}|$ .

---

### 4.2.2. Javító görbe

Tegyük fel, hogy egy  $G$  görbekövetés a MEGÁLLÁSZÁRT( $p_0, p_i, \varepsilon$ ) feltétellel megállt, ekkor megpróbáljuk a görbét bezárni. Ehhez elindítunk egy  $G_j$   $\frac{p_0+p_i}{2}$ -ből kiinduló kétirányú zárt vonalkövetést MEGÁLLÁSDUPLAZÁRT( $q_i, p_i, \varepsilon$ ) megállási feltétellel. Amennyiben  $G_j$  gond nélkül létrejött a megadott megállási feltétellel, a kívánt zárt görbe létrehozható a  $G$  és  $G_j$  görbék megfelelő átlagolásával.

### 4.2.3. Görbék átlagolása

A zárt görbe létrehozásához átlagolnunk kell a  $G$  és  $G_j$  görbét. Ezt olyan súlyozással tesszük, hogy egy görbének mindig a „közepe” domináljon és a szélső pontjait elhanyagoljuk. Mivel a  $G$  paraméterezett görbe körbe ér, csak nem zárt,  $G$  mentén egyenletesen átlagolhatjuk a görbét.

Legyen  $LK(p, G_j)$  a  $G_j$  görbe  $p$ -hez legközelebbi pontja,  $d$  a felosztás finomsága, és az algoritmus kimenete a  $G_a$  átlagolt görbe. Legyen  $f : [0, g] \rightarrow [0, 1]$  súlyfüggvény, ahol  $g = hossz(G)$ .

$$f(x) := \begin{cases} \frac{2x}{g} & \text{ha } 0 \leq x < \frac{g}{2} \\ 1 - \frac{2x-g}{g} & \text{ha } \frac{g}{2} \leq x \leq g \end{cases}$$

---

#### Algoritmus 7 ZÁRTGÖRBE( $G, G_j$ )

---

```

 $p = LK(G(0), G_j)$ 
for  $x := d \rightarrow hossz(G)$  step  $d$  do
   $V(G_a) \leftarrow p$ 
   $q = LK(G(x), G_j)$ 
   $p_n = f(x)G(x) + (1 - f(x))q$ 
   $e = (p, p_n)$ 
   $E(G_a) \leftarrow e$ 
   $p = p_n$ 
end for

```

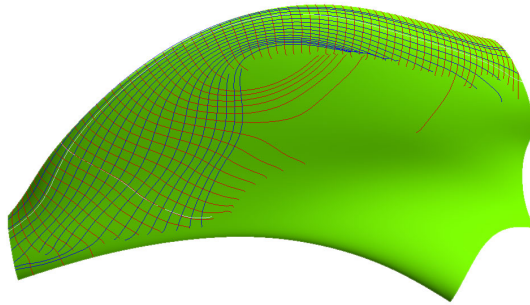
---

Az algoritmus végeredménye tehát egy  $G$ -re és  $G_j$ -re is jól illeszkedő zárt görbe a felületen.

## 5. Görbületi rácsok

A felületeken végigkövetett görbületi vonalak segítségével létre tudunk hozni a felületen ún. görbületi rácsokat (11. ábra). *Görbületi rácsnak* nevezünk egy négyzetrácsszerűen elhelyezkedő görbületi vonalsereget, amely egy görbületi vektormezőn követett  $R_v = \{v_1, v_2, \dots, v_n\}$  és a rá merőleges vektormezőn követett  $R_h = \{h_1, h_2, \dots, h_k\}$  görbékéből áll. Egy  $v_i$  görbét az  $R_h$ -beli görbék görbületarányosan, egyenletesen metszenek el, mivel a görbületi rácsok elvárt tulajdonsága, hogy a görbületi skalármező szerint egyenletesek legyenek, azaz a nagyobb görbületű részekben sűrűbbek, a kisebbekben ritkábbak. A görbületi rács lokálisan mindenhol létezik, de egy egységes összefüggő görbületi rács nem mindig hozható létre, például a felületen található umbilikus pontok miatt. Szintén gondot okozhatnak a lyukak és a gyenge görbületű tartományok is.

Ezeket a rácsokat széles körben használhatjuk különböző feladatokra, például görbületi rácsokkal könnyen megkaphatjuk egy felületdarab görbület szerinti paraméterezését. Ahol létrehozható egy görbületi rács, logikailag összefüggőnek tekinthetjük, így ezek szegmentációs feladatokhoz is jól használhatóak.



11. ábra. Görbületi rács egy szabadformájú felületen

A söpört felületek felismeréséhez szintén görbületi rácsokat használunk, ahol lényegében azt mondhatjuk, hogy az  $R_v$  görbék a profilt, az  $R_h$  görbék pedig a vezérgörbét adják. Később ezt pontosítjuk, és láthatjuk, hogy hogyan kapunk görbületi vonalából profilt, illetve vezérgörbét.

### 5.1. Görbületarányos paraméterezés

A görbületi rácsok létrehozásakor azt a szemléletet követtük, hogy a nagyobb görbületű régiókon legyen a rács sűrűbb, míg kisebb görbületű régiókon ritkább legyen. A módszer lényege, hogy a vonalkövetés során a görbéinket görbületarányosan paraméterezzük fel, majd a rá merőleges irányban a követett görbét már a paraméterezés szerint egyenletesen indítjuk.

A vonalkövetés közben a görbületi vonalakat görbületi skalármező segítségével paramétereztük, ami konkrétan azt jelenti, hogy a görbe egy  $e$  szakaszának a hosszát beszoroztuk az ott becsült  $\kappa(e) + c$  paraméterrel ahol  $\kappa(e)$  a becsült felület  $e$  iránymenti normálgörbülete,  $c$  a korrekciós konstans.

A  $c$  paraméterre azért van szükség, mert közel síkszerű felületen a görbület közel 0, így ott túl ritka lenne a görbületi rács, és azért is hasznos, mert a zajosságból eredő görbületi bizonytalanságok aránya kevésbé hat a paraméterezésre.

$$\text{hossz}(e) = |e|(c + \kappa(e))$$

## 5.2. Naiv algoritmus

Naiv algoritmusnak nevezzük azt, amikor egy  $p_0$  kiindulópontból felvesszünk két egymásra merőleges  $V$  és  $H$  görbületi vonalakat, majd ezek mentén egyenletesen az  $R_v$  és  $R_h$  görbéket amik megadják a görbületi rácsot. Ez az algoritmus lokálisan egy jó görbületi rácsot ad (azaz abban az esetben ha a  $V$  és a  $H$  nem túl hosszú), de lyukakban, gyenge görbületű régiókban megakad, és a  $p_0$ -tól messzebb lévő területeken már nem lesz jó a rács. Legyen  $d_h, d_v$  a felosztás finomsága,  $\text{TRACE}(p, \kappa)$  egy  $p$ -ből induló kétirányú vonalkövetés  $\kappa$  vektormezőn,  $\kappa_o$  pedig a  $\kappa$ -ra merőleges vektormező.

---

### Algoritmus 8 NAIVRÁCS( $p_0, \kappa, d_h, d_v$ )

---

```

 $V = \text{TRACE}(p_0, \kappa)$ 
 $H = \text{TRACE}(p_0, \kappa_o)$ 
for  $x := 0 \rightarrow \text{hossz}(V)$  step  $d_h$  do
   $R_h \leftarrow \text{TRACE}(V(x), \kappa_o)$ 
end for
for  $x := 0 \rightarrow \text{hossz}(H)$  step  $d_v$  do
   $R_v \leftarrow \text{TRACE}(H(x), \kappa)$ 
end for

```

---

## 5.3. Lyukak kikerülése

Fel kell készülnünk arra, hogy a felületeken lyukak találhatóak. Ezen belül azzal az esettel foglalkoztunk, mikor a söpört felületen nincsenek lyukak egymással szemben, azaz hogy a söprő sík által kimetszett görbe mindig összefüggő. Erre az esetre adtunk egy megoldást (12. ábra).

A módszer lényege, hogy egy hosszanti görbületi vonallal megyünk amíg tudunk, majd ha elakadunk, a végén egy rá merőleges vonalon egyenletesen felvesszünk hosszanti görbéket, és a leghosszabbal folytatjuk a rács elkészítését.

Legyen  $\kappa$  vezérgörbe irányú görbületi vektormező,  $\kappa_o$  rá merőleges. Kiindulási pont a  $p_0$ , és  $d_v, d_h$  a felosztás finomsága.  $\text{TRACES}(p, \kappa)$  egy  $p$ -ből induló egyirányú,  $\text{TRACED}(p, \kappa)$  kétirányú vonalkövetés  $\kappa$  vektormezőn, ami ha kell zárt görbéket hoz létre.

**Algoritmus 9** LYUKASSÖPÖRTRÁCS( $p_0, \kappa, d_h, d_v$ )

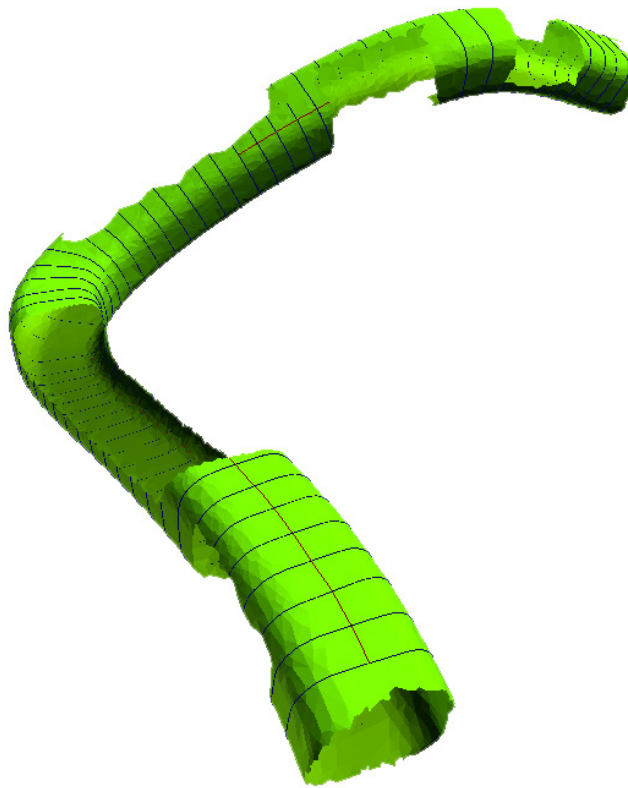
---

```

 $V = \text{TRACES}(p_0, \kappa)$ 
 $H = \text{TRACED}(p_0, \kappa_o)$ 
 $R_v \leftarrow V$ 
while hossz(LEGHOSSZABB( $R_v$ )) > d do
   $V = \text{LEGHOSSZABB}(R_v)$ 
  for  $x = 0 \rightarrow \text{hossz}(V)$  step  $d_h$  do
     $R_h \leftarrow \text{TRACED}(V(x), \kappa_o)$ 
  end for
   $H = \text{UTOLSO}(R_h)$ 
   $R_v = \emptyset$ 
  for  $x := 0 \rightarrow \text{hossz}(H)$  step  $d_v$  do
     $R_v \leftarrow \text{TRACES}(H(x), \kappa)$ 
  end for
end while

```

---



12. ábra. A LYUKASSÖPÖRTRÁCS algoritmus eredménye egy lyukas söpört felületen

## 6. A profil- és vezérgörbe meghatározása

Az egyszerűség kedvéért most tegyük fel, hogy a felület valóban egy definíció szerinti söpört felület, lyukak, és zajos régiók nélkül. Lyukak és gyenge görbületű régiók jelenlétekor az előző fejezetben ismertetett LYUKASSÖPÖRTRÁCS segítségével is meg lehet határozni a profilt, majd később a vezérgörbét is. Ahhoz, hogy ezt megtegyük, először az egységes görbületi vektormező mentén kövessünk le egy főgörbületi vonalat, majd a vonal mentén egyenletesen vegyünk fel a profillal párhuzamos, a vezérgörbére merőleges görbületi vonalakat.

Felmerülő probléma, hogy bár feltettük, hogy létezik egységes vektormező, még nem biztos, hogy azon a merőlegesen követett görbületi vonalak síkszerűek lesznek, mivel tipikusan mért adatokkal dolgozunk, csak egy bizonyos tolerancián belül beszélhetünk síkszerűségről. Ezek után kiválasztva a síkszerű görbét elsődlegesen fedésbe hozzuk őket a vektormezővel párhuzamos főgörbületi vonal iránya és a görbék kiindulópontjában becsült felületi normális szerint. Látni fogjuk, hogy ez a fedésbe hozás egyáltalán nem kielégítő, sőt még számítógéppel előállított tesztmodelleken sem elegendő. Ezek után a görbéket az ismert ICP eljárással [7] pontosabban fedésbe hozzuk, majd ha elegendően közel kerültek egymáshoz, egyesítjük őket. Az így kapott görbe egy jó profiljelölt lesz, ami további vizsgálatokkal eldönthető, hogy megfelelő lesz-e profilnak.

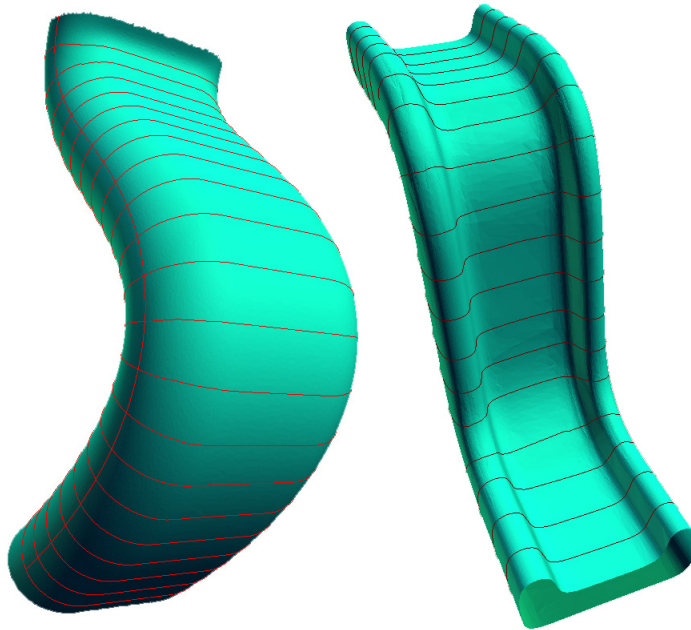
A módszer menete tehát vázlatosan:

1. Egy hosszanti főgörbületi vonal felvétele, azon egyenletesen rá merőleges nyílt vagy zárt görbületi vonalak felvétele
2. Görbületi vonalak síkszerűségének vizsgálata, a legjobbak megtartása
3. Görbék fedésbe hozása a kiindulógörbe, és a felületi normálisok alapján
4. Görbék pontos fedésbe hozása az ICP eljárás segítségével
5. Görbék egyesítése, profiljelölt meghatározása
6. Profiljelölt visszaillesztése, a vezérgörbe és a rotációs függvény meghatározása
7. A rekonstruált söpört felület vizsgálata, az eredmények kiértékelése

### 6.1. Görbületi vonalak

Először végigkövetünk egy kétirányú vektormezővel párhuzamos görbét, de megfelel az egyirányú görbekövetés is, ha az a söpört felület elejéről indul. Ezek után ezen megfelelő paraméterezés szerint egyenletesen veszünk fel merőleges görbületi vonalakat, ahogy azt a 13. ábrán láthatjuk. Így megkapjuk a fedésbe hozandó profiljelöltjeinket.

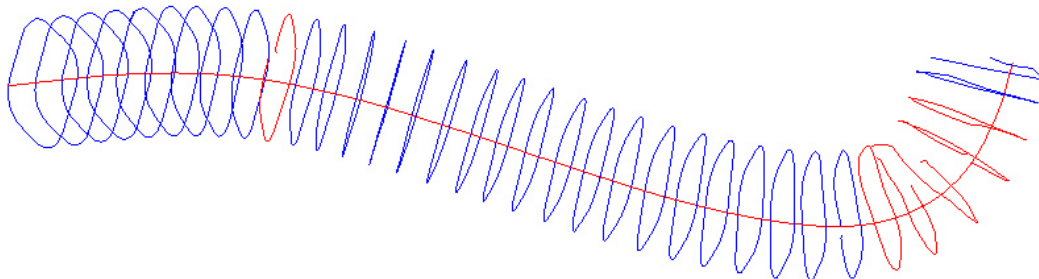




13. ábra. Vonalkövetés: főgörbe, és a rá merőleges profiljelöltek

A görbék értékelniük kell síkszerűségük szerint. Ehhez először a profilokhoz készítünk egy sík-illesztést azzal a kényszerrel, hogy a pontok átlaga a síkon van. Ezt a síktól való négyzetes távolság minimalizálásával tesszük, ami visszavezethető egy sajátvektor problémára (lásd 2.3. fejezet), így a normálvektor könnyen megkapható. Ezek után a görbe paraméterezése szerint egyenletesen elhelyezkedő pontok illesztett síktól való négyzetes távolságának átlagát vesszük, amit síkszerűségi mutatónak fogunk használni, aminek az eredményét a 14. ábrán szemléltetünk.

Ha a felület lyukas, és a LYUKASSÖPÖRTRÁCS algoritmussal kaptunk görbületi rácsot, több vezérgörbe irányú görbületi vonalunk is van a felület mentén. Ebben az esetben a következőkben ismertetett algoritmust szakaszonként kell végrehajtanunk.



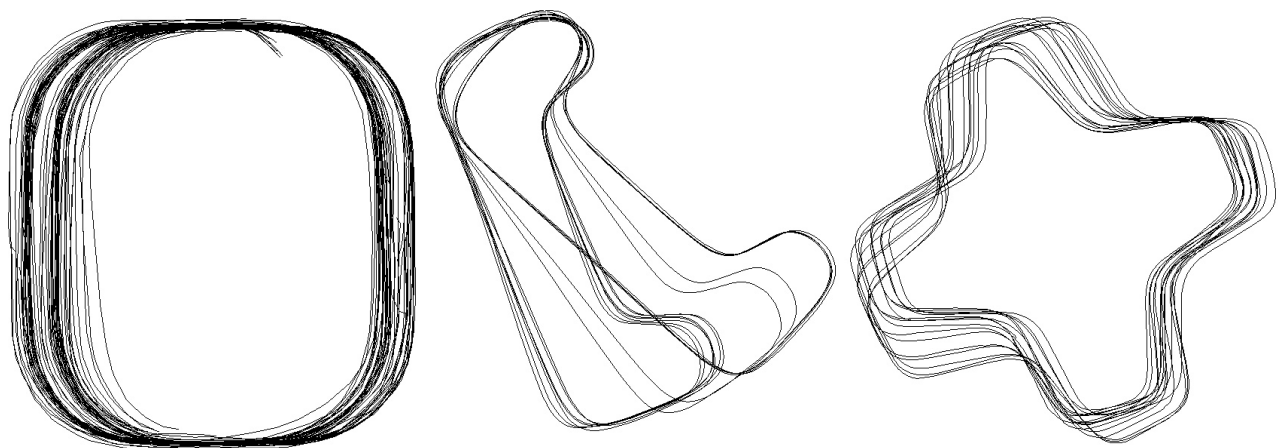
14. ábra. A piros profiljelöltek a síkszerűségi túréshatáron kívül estek

## 6.2. Elsődleges illesztés

A merőlegesen követett profiljelölteket először a kiinduló görbületi vonal, a felületi normálisok és a görbékre illesztett sík alapján próbáljuk fedésbe hozni, a következő kényszerek alapján:

- Kiindulópontjaik egy pontban legyenek
- A kiindulópontban becsült normálvektorok irányai megegyezzenek
- A görbék egy síkban legyenek, a síkillesztés alapján

Ezeket egy egyszerű eltolás, majd forgatás segítségével megkaphatjuk, így elsődlegesen illesztettük a görbéinket (15. ábra). Láthatjuk, hogy ezek mért adatok mellett nem adnak megfelelő illesztést, és hasonló felületen becsült paraméterek segítségével sem kaphatnánk lényegesen jobb eredményt. Erre csak azért van szükség, hogy a következő lépéshez használt ICP algoritmus már egy jó kiindulólélelyzetből illesszen, és így gyorsabban kaphatunk pontosabb eredményt.



15. ábra. A profiljelöltek elsődleges illesztés után

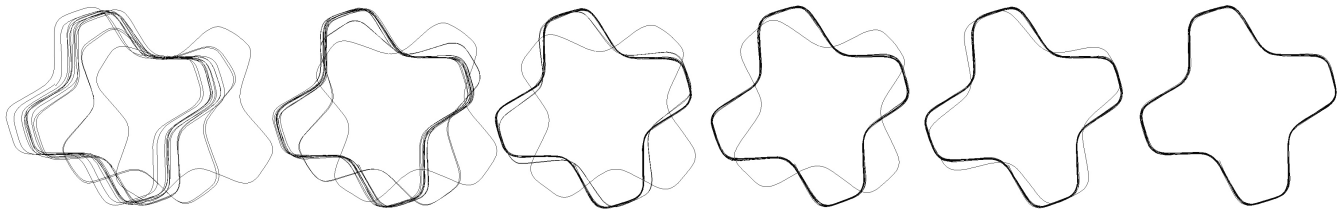
## 6.3. Pontos illesztés, az ICP eljárás

A profiljelölteket az elsődleges fedésbe hozása nem lesz kielégítő, hiszen ahogy az előbb láttuk, ez még nem ad pontos eredményt, így még nem átlagolhatjuk ki a görbéinket. Ehhez az ICP eljárást használtuk [7]. Erről nagyon sok részletes szakirodalom található, ezért csak a lényegét foglaljuk össze.

Adott két,  $A$  és  $B$  ponthalmaz, és ezeknek keressük a legjobb illeszkedését. Először minden  $a \in A$  ponthoz megkeressük a hozzá legközelebb eső  $cl(a) \in B$  bel pontot. Ezek után megkeressük a legjobb olyan eltolást és forgatást amivel az  $A$  ponthalmazt transzformálva, egy olyan ponthalmazt kapunk amivel ő és az  $A$  pontjaihoz legközelebb eső  $B$  bel pontok ( $cl(A)$ ) átlagos távolsága minimális. A transzformációt  $A$ -ra alkalmazva az algoritmust újra végrehajtjuk a következő iterációs

lépésben. Ennek van értelme, hiszen minden lépésben általában más  $B$ -beli pontok lesznek a legközelebb  $A$  pontjaihoz. Az iteráció a ponthalmazok legjobb illesztéséhez fog konvergálni, amit a 16. ábrán illusztrálunk.

Az egyes iterációs lépések meghatározásához SVD felbontásos módszert alkalmaztunk, aminek a lényege, hogy megkeressük a legjobb transzformációt ami egy tetszőleges lineáris transzformáció lehet, majd ebből csak a forgatást és az eltolást tartjuk meg. Ez egyes lépésekben kevésbé ad pontos eredményt, mintha tisztán kiszámolnánk a legjobb egybevágósági transzformációt, viszont jóval gyorsabban számítható, és a megfelelő számú iterációval ez a pontatlanság már nem számít [6].



16. ábra. A görbék fedésbe hozása az ICP eljárás segítségével, a képek 0, 2, 4, 6, 8, 10 iteráció után készültek

Az ICP algoritmus tehát megadja két ponthalmaz egy jó fedésbe hozását a legközelebbi pontok négyzetes távolságának minimalizálásával. A görbéinket úgy hozzuk fedésbe, hogy a görbék paraméterezése szerint egy  $d$  távolsággal egyenletesen felvesszünk pontokat, majd ezekkel elvégezzük a megfelelő számú ICP iterációt. Ezt úgy tesszük meg, hogy kiválasztjuk a leghosszabb görbék közül a legsíkszerűbbet (feltételezzük, hogy ez a legjobb görbe), majd az összes többi görbét ehhez igazítjuk.

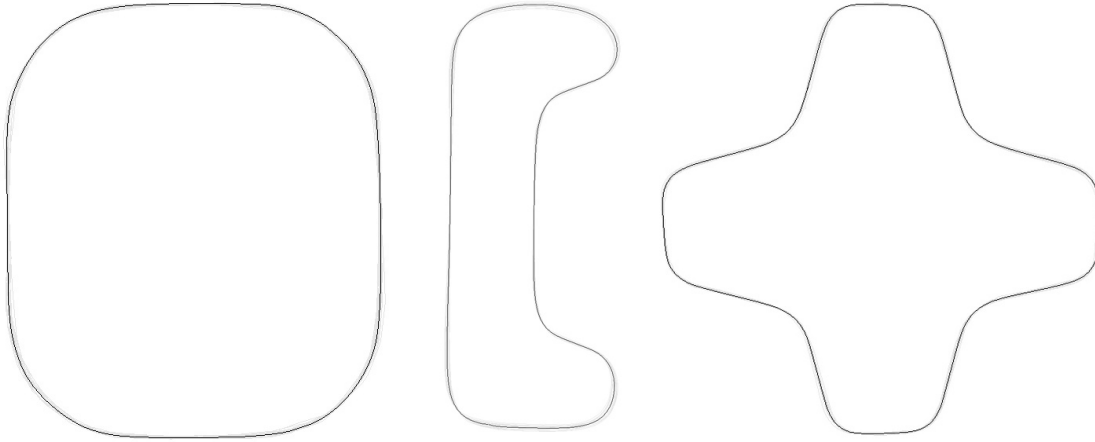
#### 6.4. A görbék átlagolása

Az ICP eljárás után a görbék átlagolásával kaphatunk egy optimalizált görbét, amelyet egy jó profiljelöltnek tekinthetünk. Az átlagolásra két különböző módszert is kipróbáltunk.

1. A középpontból körbeforgó félsík mentén átlagoljuk a síkkal való metszéspontokat
2. Egy profiljelöltön egy síkkal végigsöpörve csak egy bizonyos közelségben levő pontokat átlagoljuk

Az 1. megoldás csak csillagszerű alakzatokon működik jól, mivel ha egy félsík több helyen is metsz egy profilt, rossz eredményt kapunk. Az 1. módszer végeredményét a 17. ábra első és harmadik görbéjén szemléltetjük.

A 2. megoldás éles töréseknél ad néha rosszabb eredményt, mert ott bizonytalan a normálvektor becslés, és a söprő sík nem mindig áll megfelelő helyzetben. A 2. módszer eredményét a 17. ábra második görbéjén láthatjuk.



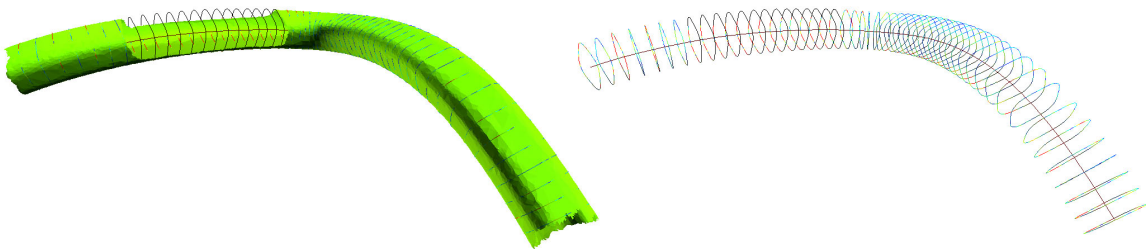
17. ábra. Átlagolt profiljelöltek

### 6.5. A vezérgörbe meghatározása

A teljes digitális reprezentáció meghatározásához szükségünk van a vezérgörbe, és a rotációs függvény meghatározására is. Ehhez a profilmeghatározás közben használt igazításokat használjuk fel.

A profil meghatározásához a görbületi vonalakat fedésbe hoztuk, átlagoltuk és így megkaptuk a profilt. Az egyes görbületi vonalakhoz tehát megvan, hogy azt a felületről milyen transzformáció hozza legjobban fedésbe a becsült profillal. Ennek a transzformációnak az inverzével pedig a profil minden egyes görbéhez visszahelyezhető a felületre, így a profil mindenhol lokálisan illeszkedni fog az aktuális görbületi vonalhoz. Ha megfelelően sűrűn vesszük fel a görbületi rácsot, a visszahelyezett profilgörbe egy adott pontja mentén becsülhetünk vezérgörbét is.

Egy becsült  $P$  profilgörbe meghatározásához a  $G_i$  görbületi vonalakat fedésbe hoztuk, majd átlagoltuk. A  $G_i$ -hez tartozó transzformáció legyen  $T(G_i)$ . Ekkor a becsült profilt a felületre minden  $G_i$ -hez visszailleszthetjük, így  $T(G_i)^{-1}P$  görbe a felületre is jól illeszkedik (18. árba). Ezzel a vezérgörbe néhány pontja is meghatározható, legyen  $v \in \mathbb{R}$  tetszőleges, ekkor  $p_i = T(G_i)v$  pontok megadják a vezérgörbe pontjait, amire görbét illeszthetünk, vagy megelégedhetünk a  $\{p_1, p_2, \dots\}$  törtvonallal is, ha a pontok elég sűrűn helyezkednek el. A rotációs függvény diszkrét pontjait pedig a  $r_i = T(G_i)^{-1}(1, 0, 0)^T$  pontok adják.



18. ábra. A visszahelyezett profilgörbék, és a becsült vezérgörbe egy söpört felületen

## 7. Az eredmények kiértékelése

A digitális alakzatrekonstrukcióban fontos, hogy az illesztett felület esetében mérni tudjuk, hogy az illesztés mennyire megfelelő. Ha az érték határértéken belül van mondhatjuk, hogy megkaptuk a felület egy jó matematikai reprezentációját. Esetünkben először érdemes vizsgálni, hogy az átlagolt profilgörbe milyen jól illeszkedik a felületre különböző pontokban, mekkora a legnagyobb távolság, mennyi az átlagos eltérés és hasonlók. A teljes reprezentációt az ún. távolság térképpel vizsgálhatjuk, ami az illesztett alakzat pontjaiban megadja, hogy az milyen messze van az eredeti háromszöghálótól, amely mért adatokból származik.

### 7.1. A profilgörbe vizsgálata

Legyen a  $\gamma$  profilgörbe törtvonal a  $\{e_1, e_2, \dots, e_n\}$  élekkel reprezentálva. A profilgörbe felületre való visszahelyezése után a görbe egy  $p$  pontjában megnézzük, hogy az milyen messze van a felülettől. Megvizsgáljuk a pontok átlagos távolságát, a legnagyobb és legkisebb eltérést. Legyen  $TÁVOLSÁG(e, M)$  az  $e$  él vagy pont  $M$  eredeti modelltől vett távolsága.

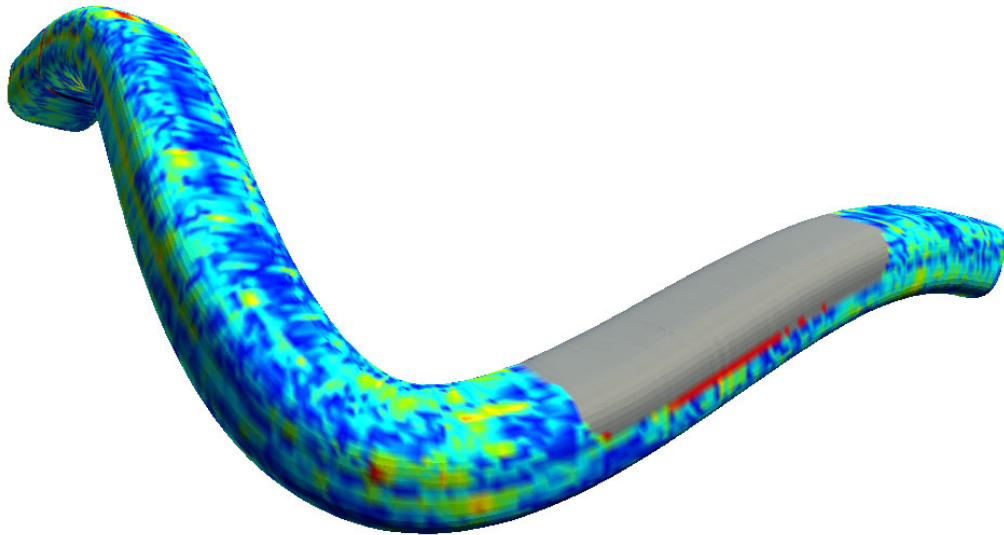
$$\text{ÁTLAGD}(\gamma, M) = \sum_{i=1}^n \frac{\text{hossz}(e_i) \text{TÁVOLSÁG}(e_i, M)}{\text{hossz}(\gamma)} \approx \int_{\gamma} \frac{\text{TÁVOLSÁG}(x, M)}{\text{hossz}(\gamma)} dx$$

$$\text{MAXD}(\gamma, M) = \max_{i \in \{1, \dots, n\}} \text{TÁVOLSÁG}(e_i, M)$$

### 7.2. Távolság térkép

Gyakran vizsgálják a rekonstruált modell és az eredeti modell viszonyát az ún. távolság térkép (deviation map) segítségével, ami az illesztett felület minden pontjában megadja az eredeti modelltől való távolságát (19. árba). Ez önmagában is érdekes lehet, hiszen az előzőekhez hasonlóan vizsgálhatunk átlagos távolságot, maximális távolságot. Nagy előnye viszont, hogy megjelenítve vizuálisan is ellenőrizhetjük az illesztést, grafikusan illusztrálva a megfelelő, illetve a toleranciát nem kielégítő részeket. Legyen  $p$  az illesztett felület egy pontja.

$$\text{DEVMAP}(p, M) = \text{TÁVOLSÁG}(p, M)$$



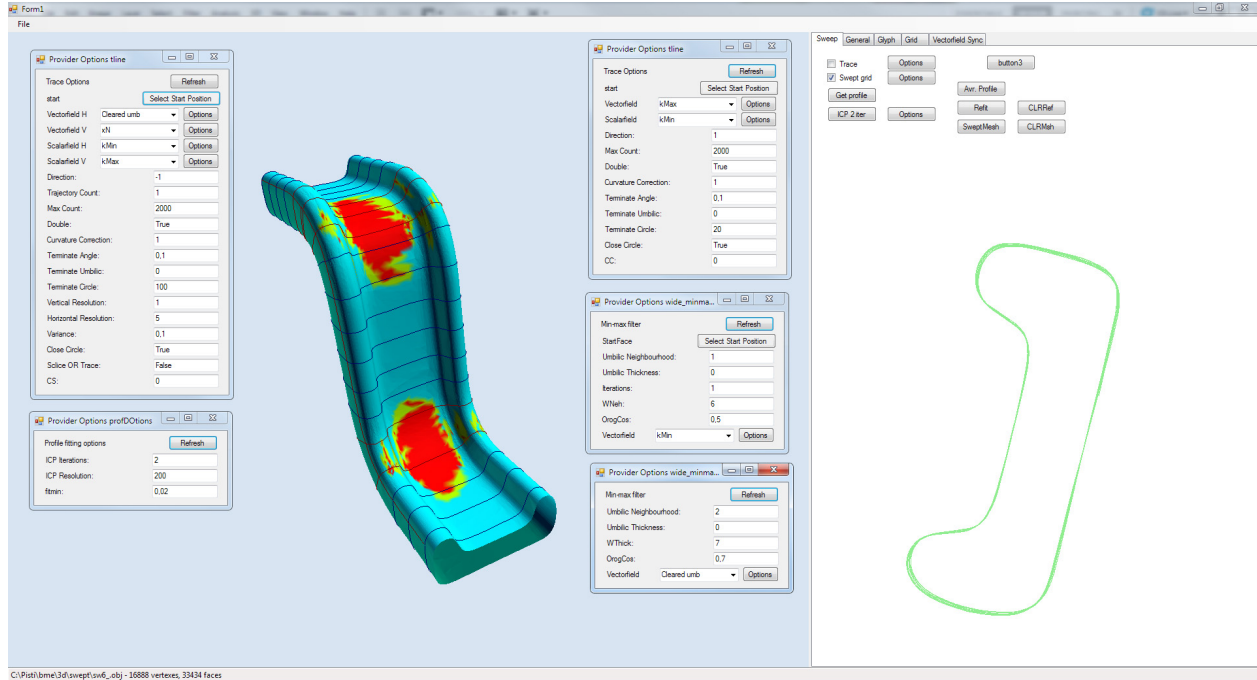
19. ábra. A távolság térkép egy illesztett söpört felületen

## 8. Megvalósítás, keretrendszer

Az ismertett algoritmusokat egy általam fejlesztett algoritmus-tesztkörnyezetben implementáltam. A tesztkörnyezetben lehetőségünk van nagy méretű háromszöghálókat beolvasni (PLY és OBJ fileípusokból), azokat fél-él adatstruktúrában bejárni, és a megfelelő görbületi jellemzőket megbecsülni. Az algoritmusok eredménye kiértékelhető a modellek színezésével vagy hisztogrammok segítségével. A tesztkörnyezet alkalmas még vektormezők, valamint a rajtuk követett görbületi vonalak és görbületi rácsok megjelenítésére is.

A program nagy előnye, hogy a megjelenítés, cache-elés, kiértékelés automatikusan történik, és csak az algoritmusnak megfelelő metódusokat kell implementálnunk. A paramétereizhetőség szintén hatékony, ezt is automatikusan kezeli a program metódusonként, és létrehozza a megfelelő bemeneti vezérlőket a képernyőn.

A program C# nyelven íródott, a megjelenítés XNA környezetben valósult meg.



20. ábra. Az általam fejlesztett algoritmus-tesztkörnyezet futás közben

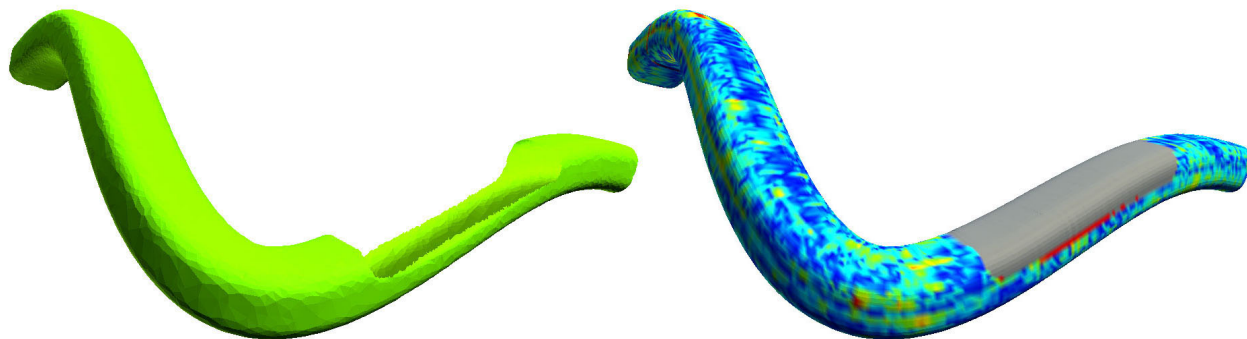
## 9. Összefoglalás, további célok

Megoldást adtunk mért adatokból származó állandó profilú söpört felületek matematikai reprezentációjának meghatározására. Ezt lokálisan becsült görbületi jellemzők segítségével oldottuk meg. Megvizsgáltuk, hogy az algoritmus kimenete mennyire jól illeszkedik az eredeti felületre, amivel eldönthető, hogy a reprezentáció milyen mértékben fogadható el. A prototípus implementáció gyorsan és hatékonyan működött több nagyméretű tesztmodellel, amelyek valós mért adatok alapján készültek. Ily módon – reményeink szerint – az algoritmus alkalmas arra, hogy söpört felületeket megbízhatóan rekonstruálni tudjunk a digitális alakzatrekonstrukció során (21., 22., 23., 24., 25. ábra).

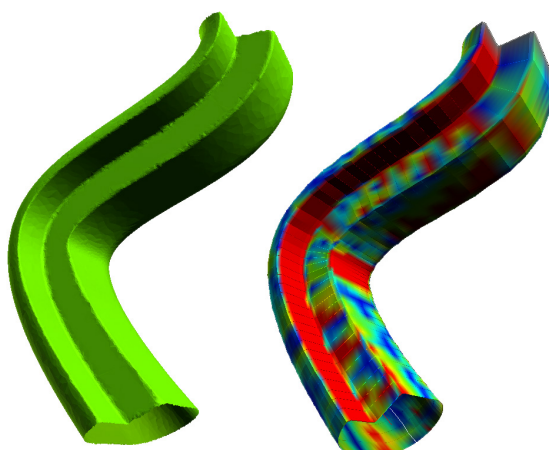
További cél lehet, hogy megvizsgáljuk és rekonstruáljuk a változó profilú söpört (lofted) felületeket, ahol meg van még adva egy skálázási függvény is, ami szerint söprés közben a profil átmérője is változik.

## Köszönetnyilvánítás

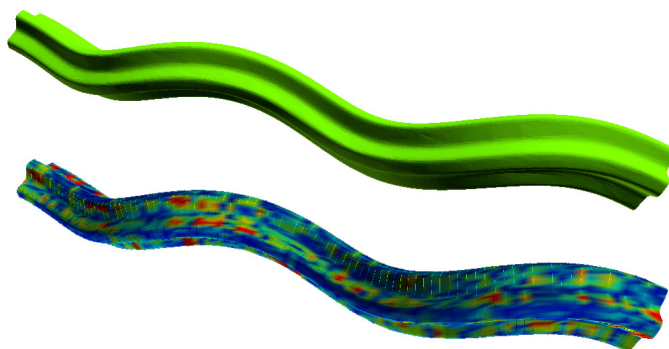
Köszönöm témavezetőmnek Várady Tamásnak a munkám irányítását, a felmerülő nehézségekben nyújtott segítséget, ötleteket és a rám fordított időt. Szintén köszönettel tartozom még Salvi Péternek a program fejlesztése során felmerülő technikai és elméleti problémák megoldása során adott segítségért.



21. ábra. Teszt 1., söpört felület illesztése

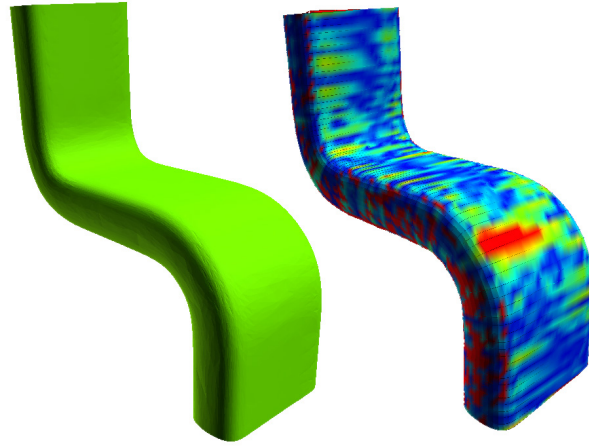


22. ábra. Teszt 2., söpört felület illesztése

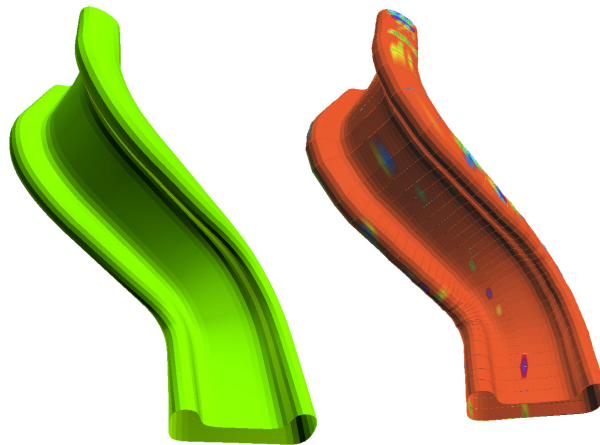


23. ábra. Teszt 3., söpört felület illesztése





24. ábra. Teszt 4., söpört felület illesztése



25. ábra. Teszt 5., söpört felület illesztése

## Hivatkozások

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2nd Edition, (1992)
- [2] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*, Academic Press, 5th Edition, (2002)
- [3] T. Várady, R. Martin, *Reverse Engineering*, G. Farin, J. Hoschek, M. S. Kim, Handbook of Computer Aided Geometric Design, Chapter 26, Elsevier, (2002)
- [4] P. Benkő, T. Várady, *Segmentation Methods for Smooth Point Regions of Conventional Engineering Objects*. In: Computer-Aided Design, **36**, Elsevier, (2004)
- [5] Zs. Terék, *Curve Networks to Segment Polygonal Meshes for Digital Shape Reconstruction*. In: Periodica Polytechnica — Electrical Engineering, **55**, (2010)
- [6] D.W. Eggert, A. Lorusso, R.B. Fisher: Estimating 3-D rigid body transformations: a comparison of four major algorithms, *Machine Vision and Applications* **9**, (1997), 272-290.
- [7] Z. Zhang: Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision* **13**, (1994), 119-152.
- [8] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, M. Desbrun, *Anisotropic polygonal remeshing*, ACM Transactions on Graphics (TOG), **22**, (2003), 485-493
- [9] T. Várady, P. Salvi, *3D-s számítógépes geometria és alakzatrekonstrukció tárgy diasorok*, BME IIT (2012)